I HEREBY CERTIFY THAT THIS CORRESPONDENCE IS BEING DEPOSITED WITH THE UNITED STATES POSTAL SERVICE AS FIRST CLASS MAIL IN AN ENVELOPE ADDRESSED TO: COMMISSIONER FOR PATENTS, P.O. BOX 1450, ALEXANDRIA, VA 22313-1450, ON THE DATE INDICATED BELOW.

BY: Clark Jablon Date: April 30, 2008

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Named Inventor:
Richard D. Martin

Conf. No.:	9008	Group Art Unit:	2154
Appln. No.:	09/923,923	Examiner:	Mohammad A. Siddiqi
Filing Date:	August 7, 2001	Attorney Docket No.:	401-13U1

Title: SYNDICATION METHODOLOGY TO DYNAMICALLY PLACE DIGITAL ASSETS ON NON-RELATED WEB SITES

SUPPLEMENTAL DECLARATION OF PRIOR INVENTION TO OVERCOME CITED PATENT (37 C.F.R. § 1.131)

This supplemental declaration is being submitted to establish completion of the invention in this application in the United States at a data prior to July 15, 2000, which is the earliest effective date of the prior art U.S. Patent No. 6,826,594 (Pettersen), which was cited and applied by the Examiner in an Office Action dated October 31, 2007.

This supplemental declaration supplements the "Declaration of Prior Invention…" submitted January 4, 2007, hereafter, the "original Declaration."

The persons making the supplemental declaration are the inventors, and are thus qualified to submit this declaration under 37 CFR § 1.131.

FACTS AND DOCUMENTARY EVIDENCE

Documents

To establish the date of completion of the invention of this patent application, copies of the documents and supporting statements submitted as evidence in the original Declaration are incorporated herein, and thus no additional copies are enclosed.

Supporting Statements

To establish the date of completion of the invention of this patent application, copies of the supporting statements submitted as evidence in the original Declaration are incorporated herein, and thus no additional copies are enclosed.

Supplemental Supporting Statements

To further establish the date of completion of the invention of this patent application, the following additional supporting statements are provided.

The following claim chart identifies where the claim features appear in Exhibit 1 (Syndication Source Code shown in Appendix A (pages A1-A8) of the present specification) and Exhibit 2 (Document entitled "Affiliate Setup Instructions").

Claims	Exhibit 1 (Syndication Source Code shown in Appendix A) with explanatory commentary	Exhibit 2 (Document entitled "Affiliate Setup Instructions") with explanatory commentary
13. A method of constructing a web page that allows for syndication of digital assets, the method comprising:		
(a) constructing a web page; and	The code in Appendix A illustrates this step through the examination of the request (as required at the syndication server) for certain parameters (see Item 1). Item 2 describes the document.write statement containing the selected content for presentation on the web site of interest. Items 5 and 6 relate to formatting and escape character handling.	Exhibit 2 shows sequentially how marketing material was made illustrating how syndication code would be placed on a subscriber's web page. Page 1 of 3 shows a standard web page using content created in flat HTML. Page 2 of 3 shows the insertion of JavaScript code (syndication code) as a substitute for that content and its subsequent presentation in the style of

	Explanatory text: • Subscriber’s Web page is requested from an anonymous user. • Request is sent to the web server for that page. That page contains syndication code that is connecting to the syndication server. • The request message contains a token letting the Syndication Server know the referring URI. • The Referring URI is checked against a DB table that ‘knows’ the ‘content sets’ that are available and ‘who’ is allowed to make requests for them. If the referring URL and the content requested matches the validation table, a document.write statement is sent back to the requesting URL for inclusion in the web page. Construction of the content of the web page is manifested by the described response-request: a set of parameters required by the syndication server to assure that the content delivered back to the authorized view of the content is correct. This includes section and article identifiers unique to the content set requested; authentication of the content, subscriber and web URI; and the ultimate response from the server of a document.write statement containing the HTML for the	the web site of interest.

	requested content.	
(b) inserting into the web page script associated with at least one digital asset that is desired to be part of a fully rendered web page, the inserted script including code to request the content of the digital asset from a remote site when the code is executed by a browser, the code including a uniform resource identifier (URI) of the web page and a unique identifier of the content.	Items 1, 3 and 4 show code snippets related to this step. The URL shown in Item 3 relates to the URI of the web page. The String ContentID and its subparts, SectionID, ArticleID, PartID, shown in Item 1 relate to the unique identifier of the content.	See Exhibit 2, page 2 of 3
14. A method of claim 25 wherein the scripting language is JavaScript.	See Items 1, 2, and 3. The code is JavaScript.	See the references to the syndication code for the web page.
15. A method of claim 13 wherein the content is an executable file.	JavaScript, upon page loading, is an executable element.	
16. A method of claim 13 wherein the script includes a subscriber identifier and a content identifier, which, together, create the unique identifier of the content.	Item 4 illustrates these parameters. The content identifier is found at two levels. First, the content is given a GUID, of sorts, where it is unique content to the syndication server of interest. This ID is found in the request string of the subscriber URI and as well in the parameters of the code from Appendix A (this is the syndication server code).	
25. The method of claim 13 wherein the script is a scripting language.	See Items 1, 2, and 3. The code is JavaScript.	See the references to the syndication code for the web page.

17.	Same as 13	Same as 13
18.	Same as 14	Same as 14
19.	Same as 15	Same as 15
20.	Same as 16	Same as 16
26.	Same as 25	Same as 25
21.	Same as 13	Same as 13
22.	Same as 14	Same as 14
23.	Same as 15	Same as 15
24.	Same as 16	Same as 16
27.	Same as 25	Same as 25

Code snippets from Exhibits 1 and 2

Item 1. Content parameters request interpretation at server (Appendix A, page A-1):

```
        String contentID, detail;
        boolean newContent = false;
        Content cont;  //needed for lookup
        Params params = new Params(req);  //decipher the request
    String ContentID = params.contentServer //req.getParameter("Server")
         + "?db=" + params.db
         + "&SectionID=" + params.SectionID
         + "&ArticleID=" + params.ArticleID
         + "&PartID=" + params.PartID
         + "&ln=" + params.linkBase
         + "&mode=" + params.mode;
    if (params.unit != null)
         {ContentID = ContentID  + "&unit=" + params.unit
          + "&number=" + params.number;
         }
```

Item 2. document.write statement used to present content at the subscribers web site found in Appendix A, page A-2.

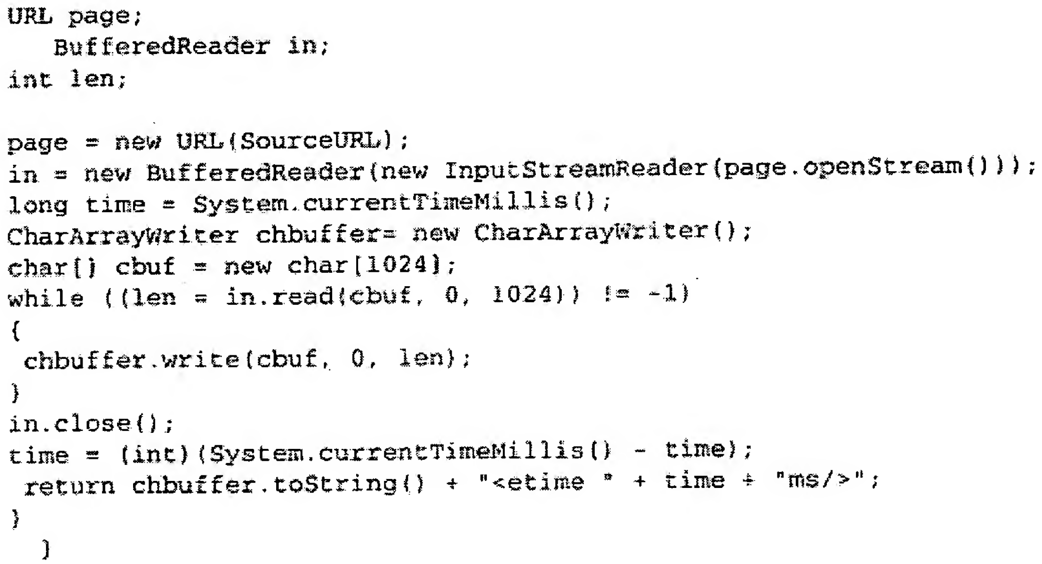
```
    public long getLastModified(HttpServletRequest req)
        {
            return lastModified/1000*1000;
        }

     private String convertTojs(String htmlText, Params params){
        htmlText = htmlText.replace('\r', ' ');
        htmlText = htmlText.replace('\n', ' ');
        htmlText = swapStrings(htmlText, "\\", backSlash);
        htmlText = swapStrings(htmlText, "\'", apostrophe);
        htmlText = swapStrings(htmlText, "\"", doubleQuote);

        if (params.ArticleID.equals("0")){
        htmlText = JavaScript.openArticle() + makeJavascriptHrefs(htmlText,
  params);
        }

        return "document.write (\""  + htmlText + "\");";
     }
```

Item 3. Code snippet in Appendix A, page A-7, that relates to presentation of a URL (URI) request.

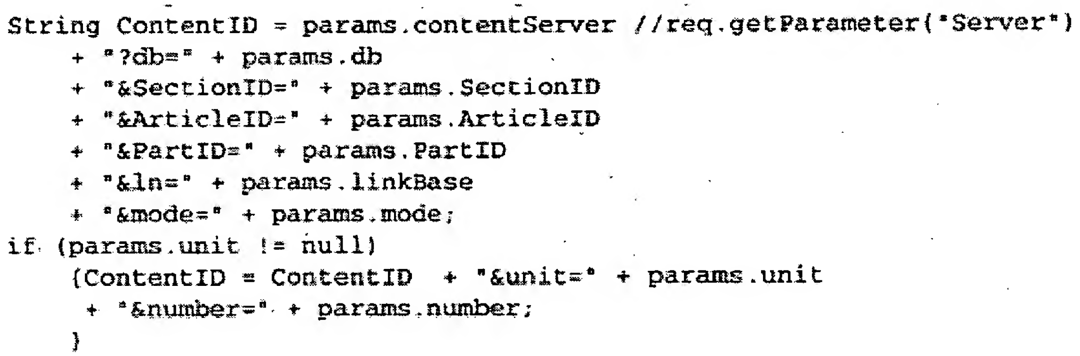


```
URL page;
   BufferedReader in;
int len;

page = new URL(SourceURL);
in = new BufferedReader(new InputStreamReader(page.openStream()));
long time = System.currentTimeMillis();
CharArrayWriter chbuffer= new CharArrayWriter();
char[] cbuf = new char[1024];
while ((len = in.read(cbuf, 0, 1024)) != -1)
{
 chbuffer.write(cbuf, 0, len);
}
in.close();
time = (int)(System.currentTimeMillis() - time);
 return chbuffer.toString() + "<etime " + time + "ms/>";
}
  }
```

Item 4. Content is unique to the syndication server and ascribed a reference ID. This is seen at the syndication server code snippet in Appendix A, page A-1.

Syndication Server Code:



```
String ContentID = params.contentServer //req.getParameter("Server")
    + "?db=" + params.db
    + "&SectionID=" + params.SectionID
    + "&ArticleID=" + params.ArticleID
    + "&PartID=" + params.PartID
    + "&ln=" + params.linkBase
    + "&mode=" + params.mode;
if (params.unit != null)
    {ContentID = ContentID  + "&unit=" + params.unit
     + "&number=" + params.number;
    }
```

<u>Item 5.</u> This section of the code in Appendix A, page A-2, eliminates interpretation problems that would typically be encountered with the use of special characters in certain publishing instances. In 1999, these were the typically found characters if interest.

```
    private String convertTojs(String htmlText, Params params){
      htmlText = htmlText.replace('\r', ' ');
      htmlText = htmlText.replace('\n', ' ');
      htmlText = swapStrings(htmlText, "\\", backSlash);
      htmlText = swapStrings(htmlText, "\'", apostrophe);
      htmlText = swapStrings(htmlText, "\"", doubleQuote);

      if (params.ArticleID.equals("0")){
      htmlText = JavaScript.openArticle() + makeJavascriptHrefs(htmlText,
  params);
      }
```

Item 6. The following code, excerpted from Appendix A, page A-3, relates to proper formatting of the content request to the web browser requesting the content.

```
javascript:expandHeader(ArticleID)

 private String makeJavascriptHrefs(String htmlText, Params params){
  //find "<a", find "href=", find "ArticleID=", find '>'
  //replace href= with javascript:expandHeader(ArticleID)
  int indexBeginA, indexHref, indexArticleID,
      ArticleIDEndsAt, indexEndA, indexClosingA;
  String ArticleID = null;
  String htmlTextLowercase = htmlText.toLowerCase();
  StringBuffer newHtml = new StringBuffer();
  int indexStart = 0;
  indexBeginA = htmlTextLowercase.indexOf("<a ", indexStart);
  indexHref = htmlTextLowercase.indexOf("href=", indexBeginA + 2);
  indexArticleID = htmlTextLowercase.indexOf("articleid=", indexHref +
5) + 10;
  indexEndA = htmlTextLowercase.indexOf(">", indexArticleID);
  indexClosingA = htmlTextLowercase.indexOf("</a>", indexEndA) + 4;
  while (indexBeginA >= 0 && indexBeginA < indexHref
      && indexHref < indexArticleID
      && indexArticleID < indexEndA
      && indexEndA < indexClosingA){
    ArticleIDEndsAt = htmlTextLowercase.indexOf("&",
indexArticleID);
    //ArticleID = 'a' + htmlTextLowercase.substring(indexArticleID,
ArticleIDEndsAt);
    ArticleID = htmlTextLowercase.substring(indexArticleID,
ArticleIDEndsAt);
    newHtml.append(htmlText.substring(indexStart,
indexHref)).append("href=");

    newHtml.append("\\\"javascript:openArticle(\\\'" +
params.templateURL + "?SectionID="
      + params.SectionID + "&ArticleID=" + ArticleID +
"&PartID=2&db=" + params.db + "\\\');\\\"");
    //newHtml.append("\\\"javascript:expandHeader(" + ArticleID +
");\\\"");
    newHtml.append(htmlText.substring(indexEndA, indexClosingA)); //
done with one anchor.

//newHtml.append(JavaScript.appendArticleMain(params.requestURL,
params.db, params.SectionID, ArticleID) );
    //new loop starts over
    indexStart = indexClosingA;
    indexBeginA = htmlTextLowercase.indexOf("<a ", indexStart);
    indexHref = htmlTextLowercase.indexOf("href=", indexBeginA);
    indexArticleID = htmlTextLowercase.indexOf("articleid=",
indexHref + 5) + 10;
    indexEndA = htmlTextLowercase.indexOf(">", indexArticleID);
    indexClosingA = htmlTextLowercase.indexOf("</a>", indexEndA) +
```

From the above-referenced documents and supporting statements, we submit that it has been established that the invention in this application was made prior to July 15, 2000, which is the earliest effective date of Pettersen.

TIME OF PRESENTATION OF THE DECLARATION

This declaration is being submitted prior to a final rejection.

DECLARATION

As a person signing below:

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Full name of sole inventor, if any Richard D. Martin

Inventor's Signature

Date 4-30-08

Residence Bethlehem, Pennsylvania

Citizenship United States of America

Post Office Address 4646 Virginia Drive, Bethlehem, Pennsylvania 18017

Full name of second joint inventor, if any John E. Wetzel

Inventor's Signature

Date 4-30-08

Residence Bath, Pennsylvania

Citizenship United States of America

Post Office Address 313 South Egg Road, Bath, Pennsylvania 18014